DWA_02.8 Knowledge Check_DWA2

---

1. What do ES5, ES6 and ES2015 mean - and what are the differences between them?

ES5, ES6, and ES2015 are different versions of the ECMAScript (ES) specification, which is a standardized scripting language specification that JavaScript follows. THE ES stands for Ecma script
ES5 was released in 2009 and brought significant enhancements to the JavaScript language. Some notable features introduced in ES5 include:
Strict mode: A stricter mode of JavaScript that enforces better coding practices and eliminates certain error-prone behaviors
ES6, also known as ES2015, was a major update to JavaScript released in 2015. It introduced numerous new features and syntax improvements, making JavaScript more powerful and expressive. Some notable features introduced in ES6 include:
Block-scoped variables: The let and const keywords for declaring variables with block scope, improving variable scoping rules.

---

2. What are JScript, ActionScript and ECMAScript - and how do they relate to JavaScript?

JScript is Microsoft's legacy dialect of the ECMAScript standard that is used in Microsoft's Internet Explorer 11 and older.
JavaScript is technically titled ECMAScript . while ActionScript, IS A JavaScript knock-off language created by Macromedia.it was initially designed for controlling simple two-dimensional vector animations made in Adobe Flash (formerly Macromedia Flash). Initially focused on animation, early versions of Flash content offered few interactivity features, thus had very limited scripting capability. Later versions added functionality allowing for the creation of web-based games and rich web applications with streaming media (such as video and audio). Today, ActionScript is suitable for

desktop and mobile development through Adobe AIR; it is used in some database applications and in basic robotics as in Make Controller Kit.

---

3. What is an example of a JavaScript specification - and where can you find it?

The specification is a collection of documents describing how JavaScript and its variants should work in the context of JavaScript and its variants. For example, here is an extract from the very first specification created by ECMA explaining how single-line commenting should work:

> *Because a single-line comment can contain any character except a "LineTerminator" character, and because of the general rule that a token is always as long as possible, a single-line comment always consists of all characters from the "//" marker to the end of the line. However, the "LineTerminator" at the end of the line is not considered to be part of the single-line comment; it is recognized separately by the lexical grammar and becomes part of the stream of input elements for the syntactic grammar. This point is very important because it implies that the presence or absence of single-line comments does not affect the process of automatic semicolon insertion*

In short:

The compiler should not execute single-line comments.
Single-line comments should always start with "//"
Single-line comments accept any character except line breaks.
Single-line comments end before the first line breaks

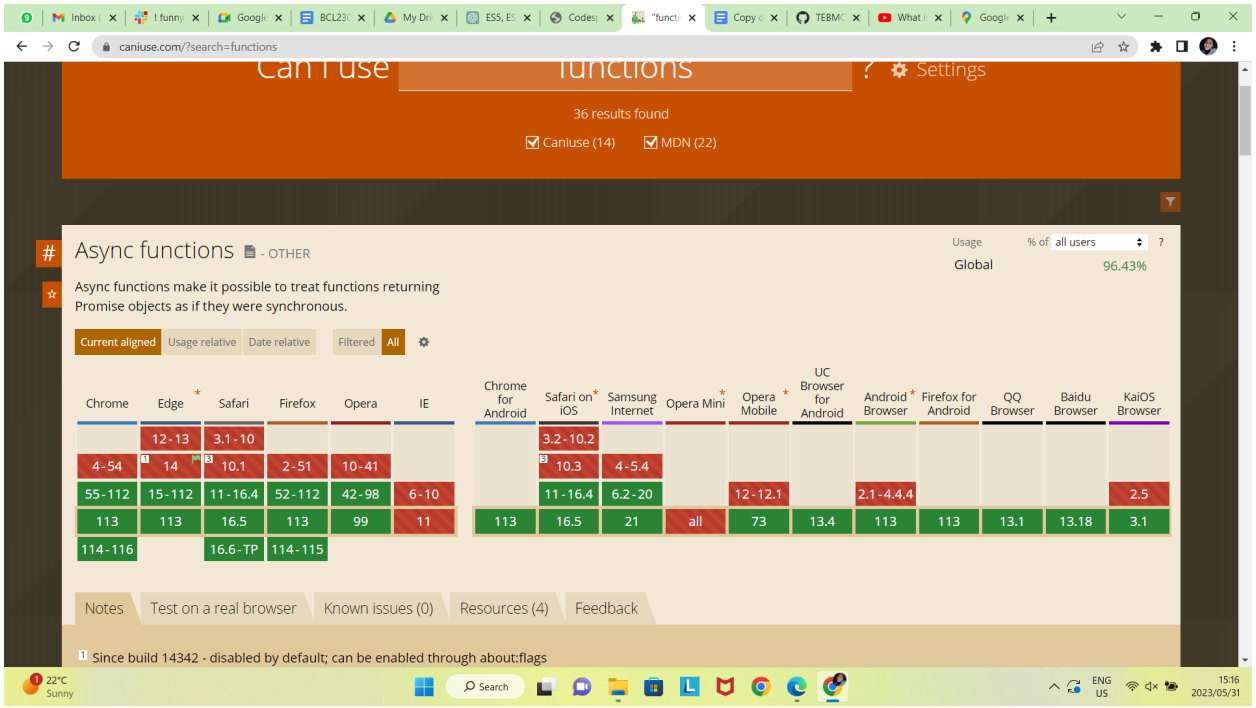
---

4. What are v8, SpiderMonkey, Chakra and Tamarin? Do they run JavaScript differently?

V8 is a free and open-source JavaScript and WebAssembly engine developed by the Chromium Project for Chromium and Google Chrome web browsers
SpiderMonkey is open-source JavaScript and WebAssembly engine by the Mozilla Foundation.[3
It is the first JavaScript engine, written by Brendan Eich at Netscape Communications, and later released as open source and currently maintained by the Mozilla Foundation. It is used in the Firefox web browser.
SpiderMonkey was released 1995. It implements the ECMA-262 specification (ECMAScript) and is written in C/C++.
Chakra is a proprietary JScript engine developed by Microsoft
Tamarin is a discontinued free software virtual machine with just-in-time compilation (JIT) support intended to implement the 4th edition of the ECMAScript (ES4) language standard. Tamarin source code originates from ActionScript Virtual Machine 2 (AVM2)

---

5. Show a practical example using **caniuse.com** and the MDN compatibility table.


caniuse.com/?search=functions
Can I use
functions
Settings
36 results found
Caniuse (14)
MDN (22)
Async functions - OTHER
Async functions make it possible to treat functions returning Promise objects as if they were synchronous.
Usage
% of all users
Global
96.43%
Current aligned
Usage relative
Date relative
Filtered
All
Chrome
Edge
Safari
Firefox
Opera
IE
Chrome for Android
Safari on iOS
Samsung Internet
Opera Mini
Opera Mobile
UC Browser for Android
Android Browser
Firefox for Android
QQ Browser
Baidu Browser
KaiOS Browser
12-13
3.1-10
3.2-10.2
4-54
14
10.1
2-51
10-41
10.3
4-5.4
55-112
15-112
11-16.4
52-112
42-98
6-10
11-16.4
6.2-20
12-12.1
2.1-4.4.4
2.5
113
113
16.5
113
99
11
113
16.5
21
all
73
13.4
113
113
13.1
13.18
3.1
114-116
16.6-TP
114-115
Notes
Test on a real browser
Known issues (0)
Resources (4)
Feedback
Since build 14342 - disabled by default; can be enabled through about:flags
22°C
Sunny
Search
ENG
US
15:16
2023/05/31